HY240: Δομές Δεδομένων
Χειμερινό Εξάμηνο – Ακαδημαϊκό Έτος 2016-2017
Διδάσκουσα: Παναγιώτα Φατούρου
Προγραμματιστική Εργασία - 1ο Μέρος

Ημερομηνία Παράδοσης: Δευτέρα, 14 Νοεμβρίου 2016, ώρα 23:59.

Τρόπος Παράδοσης: Χρησιμοποιώντας το πρόγραμμα turnin. Πληροφορίες για το πώς λειτουργεί το turnin παρέχονται στην ιστοσελίδα του μαθήματος.

Γενική Περιγραφή

Στην εργασία αυτή καλείστε να υλοποιήσετε ένα πρόγραμμα που προσομοιώνει τη λειτουργία ενός ζωολογικού πάρκου. Το πάρκο αυτό αποτελείται από ένα σύνολο οικοσυστημάτων τα οποία φιλοξενούν διάφορα είδη ζώων και είναι ανοιχτό προς τους επισκέπτες. Στο πάρκο απασχολούνται κάποιοι εργαζόμενοι στους οποίους έχει ανατεθεί η επιμέλεια κάποιων οικοσυστημάτων και η φροντίδα των ζώων που αυτά φιλοξενούν.

Αναλυτική Περιγραφή Ζητούμενης Υλοποίησης

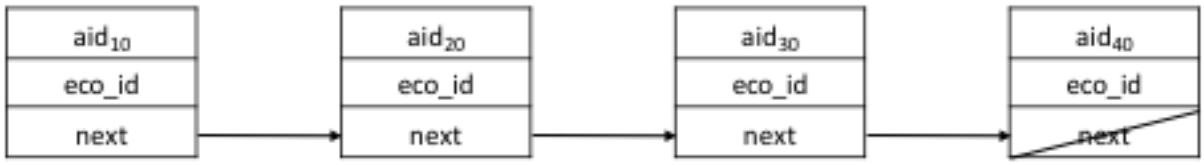
Στο ζωολογικό πάρκο κατοικούν διάφορα είδη ζώων τα οποία συνυπάρχουν ομαδοποιημένα σε οικοσυστήματα τα οποία τους επιτρέπουν να συμβιώνουν αρμονικά.

Πληροφορίες για νέα ζώα που θα φιλοξενηθούν στο πάρκο αποθηκεύονται σε μια **απλά συνδεμένη λίστα**, η οποία είναι ταξινομημένη σε αύξουσα διάταξη με βάση το αναγνωριστικό του κάθε ζώου. Η λίστα αυτή ονομάζεται **λίστα ζώων**. Ο κάθε κόμβος της λίστας αυτής αποτελεί μια εγγραφή τύπου animal με τα ακόλουθα πεδία:

- **aid:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το ζώο.
- **eco_id:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το οικοσύστημα στο οποίο κατοικεί το ζώο

- **next:** Δείκτης (τύπου animal) στον επόμενο κόμβο της λίστας ζώων.

Στο Σχήμα 1 παρουσιάζεται η λίστα όλων των ζώων του ζωολογικού πάρκου.

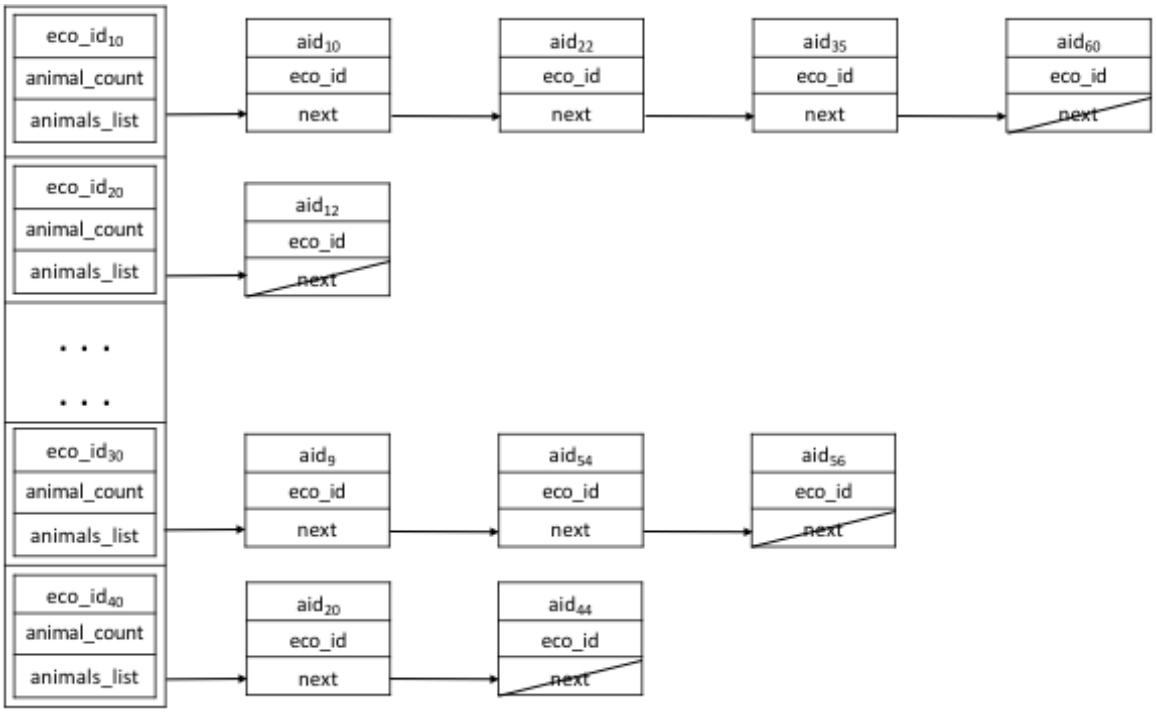


Σχήμα 1: Η απλά συνδεμένη λίστα ζώων, η οποία είναι ταξινομημένη σε αύξουσα διάταξη με βάση το αναγνωριστικό του κάθε ζώου.

Για τη διαχείριση των οικοσυστημάτων, θα δημιουργήσετε έναν πίνακα σταθερού μεγέθους **50 θέσεων** (όσα και τα οικοσυστήματα του πάρκου), ο οποίος θα ονομάζεται **πίνακας οικοσυστημάτων**. Κάθε θέση του πίνακα αποτελεί μια εγγραφή τύπου ecosystem με τα ακόλουθα πεδία:

- **eco_id:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το οικοσύστημα. Το αναγνωριστικό αυτό αποτελεί επίσης το δείκτη στη θέση του πίνακα όπου είναι αποθηκευμένο το συγκεκριμένο οικοσύστημα. Για παράδειγμα αν το πεδίου eco_id ενός κόμβου της λίστας ζώων έχει τιμή 5, αυτό σημαίνει ότι το συγκεκριμένο ζώο κατοικεί στο οικοσύστημα που βρίσκεται στη θέση 5 του πίνακα οικοσυστημάτων.
- **animal_count:** Μετρητής (τύπου int) που αντικατοπτρίζει το πλήθος των ζώων που κατοικούν στο οικοσύστημα.
- **animals_list:** Δείκτης (τύπου animal) στο πρώτο στοιχείο μιας απλά συνδεδεμένης λίστας, η οποία ονομάζεται **λίστα ζώων του οικοσυστήματος**. Η λίστα αυτή περιέχει κόμβους τύπου animal και είναι **ταξινομημένη με βάση το αναγνωριστικό των ζώων** σε αύξουσα διάταξη.

Στο Σχήμα 2 παρουσιάζεται ο πίνακας οικοσυστημάτων, και η λίστα των ζώων που δεικτοδοτείται από κάθε στοιχείο του.

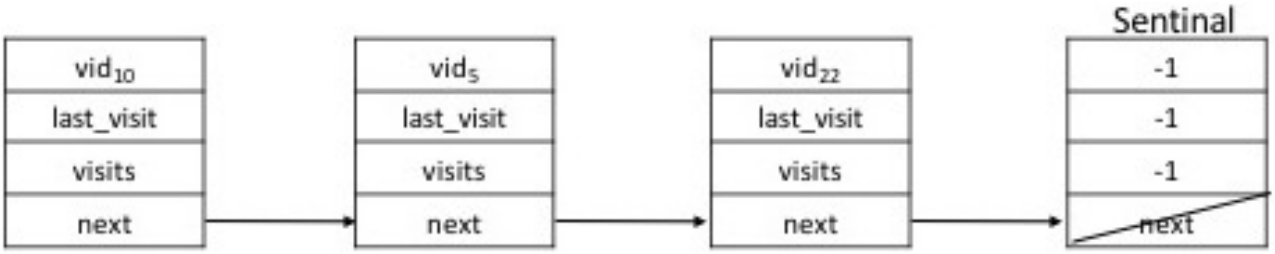


Σχήμα 2: Ο πίνακας οικοσυστημάτων, και η λίστα των ζώων που δεικτοδοτείται από κάθε στοιχείο του. Η λίστα ζώων του κάθε οικοσυστήματος είναι ταξινομημένη με βάση τα αναγνωριστικά των ζώων

Το πάρκο είναι ανοιχτό προς τους πολίτες και δεν είναι λίγοι αυτοί που το επισκέπτονται καθημερινά. Για το σκοπό αυτό θα δημιουργήσετε μια **απλά συνδεδεμένη, μη-ταξινομημένη λίστα με κόμβο φρουρό,** η οποία ονομάζεται **λίστα επισκεπτών**. Ο κάθε κόμβος της λίστας περιέχει μια εγγραφή τύπου visitor με τα ακόλουθα πεδία:

- **vid:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά τον επισκέπτη του πάρκου.
- **last_visit:** Πεδίο (τύπου int) που αντιστοιχεί στο έτος της τελευταίας επίσκεψής του στο πάρκο.
- **visits:** Μετρητής (τύπου int) που αντικατοπτρίζει τον αριθμό των επισκέψεων του επισκέπτη στο πάρκο.
- **next:** Δείκτης (τύπου visitor) στον επόμενο κόμβο της λίστας επισκεπτών.

Ο κόμβος φρουρός της λίστας είναι ένας διαχειριστικός κόμβος για τον οποίο ισχύουν τα εξής: είναι και αυτός τύπου visitor με την ιδιαιτερότητα ότι το αναγνωριστικό του (vid) και τα πεδία last_visit και visits έχουν τιμή -1 (δηλαδή δεν χρησιμοποιούνται). Το Σχήμα 3 απεικονίζει τη λίστα επισκεπτών.

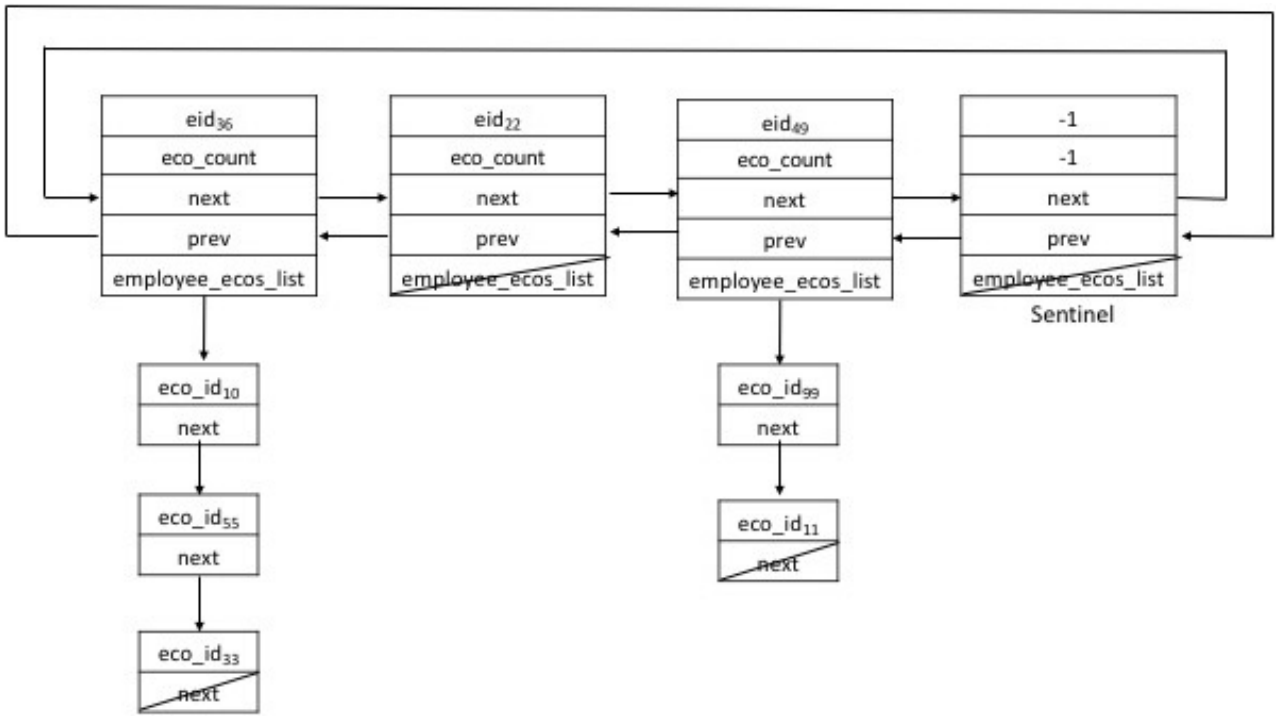


Σχήμα 3 Η απλά συνδεδεμένη, μη ταξινομημένη λίστα των επισκεπτών με κόμβο φρουρό.

Οι εργαζόμενοι στο ζωολογικό πάρκο είναι υπεύθυνοι για την επιμέλεια των οικοσυστημάτων του. Για το σκοπό αυτό θα δημιουργήσετε μια **διπλά-συνδεδεμένη, κυκλική λίστα με κόμβο φρουρό**, η οποία ονομάζεται **λίστα εργαζομένων** (employee_list). Ο κάθε κόμβος της λίστας αυτής περιέχει μια εγγραφή τύπου employee με τα ακόλουθα πεδία:

- **eid:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά τον εργαζόμενο του πάρκου.
- **eco_count:** Μετρητής (τύπου int) που αντιστοιχεί στο πλήθος των οικοσυστημάτων για τα οποία είναι υπεύθυνος ο εργαζόμενος.
- **employee_ecos_list:** Δείκτης (τύπου animal) στο πρώτο στοιχείο μιας **απλά συνδεδεμένης, μη ταξινομημένης λίστας**, η οποία ονομάζεται **λίστα οικοσυστημάτων του εργαζομένου**. Ο κάθε κόμβος της λίστας περιέχει εγγραφές τύπου employee_eco με τα ακόλουθα πεδία:
  - **eco_id:** Αναγνωριστικό (τύπου int) που χαρακτηρίζει μοναδικά το οικοσύστημα.
  - **next:** Δείκτης (τύπου employee_eco) στον επόμενο κόμβο της λίστας οικοσυστημάτων του εργαζομένου.
- **next:** Δείκτης (τύπου employee) στον επόμενο κόμβο της λίστας εργαζομένων.
- **prev:** Δείκτης (τύπου employee) στον προηγούμενο κόμβο της λίστας εργαζομένων.

Ο κόμβος φρουρός της λίστας είναι ένας διαχειριστικός κόμβος για τον οποίο ισχύουν τα εξής: είναι και αυτός τύπου employee με την ιδιαιτερότητα ότι το αναγνωριστικό του, eid, και το πεδίο eco_count έχουν τιμή -1, ενώ ο δείκτης employee_ecos_list έχει τιμή NULL. (δηλαδή δεν χρησιμοποιούνται). Επιπλέον ο δείκτης next του κόμβου φρουρού δείχνει στον πρώτο κόμβο της λίστας και το πεδίο prev του πρώτου κόμβου δείχνει στον κόμβο φρουρό ώστε η λίστα να παραμένει κυκλική. .Στο Σχήμα 4 παρουσιάζεται η λίστα εργαζομένων.



Σχήμα 4 Η λίστα εργαζομένων του ζωολογικού πάρκου. Ο κάθε κόμβος (εργαζόμενος) περιέχει μια απλά συνδεδεμένη, μη ταξινομημένη λίστα οικοσυστημάτων για τα οποία είναι υπεύθυνος ο εργαζόμενος.

Τρόπος Λειτουργίας Προγράμματος

Το πρόγραμμα που θα δημιουργηθεί θα πρέπει να εκτελείται καλώντας την ακόλουθη εντολή:

<executable> <input-file>

όπου <executable> είναι το όνομα του εκτελέσιμου αρχείου του προγράμματος (π.χ. a.out) και <input-file> είναι το όνομα ενός αρχείου εισόδου (π.χ. testfile) το οποίο περιέχει γεγονότα των ακόλουθων μορφών:

– **L <aid> <eco_id>**

Γεγονός που υποδηλώνει ότι το ζώο με αναγνωριστικό aid θα φιλοξενηθεί στο ζωολογικό πάρκο και πιο συγκεκριμένα στο οικοσύστημα με αναγνωριστικό <eco_id>. Κατά το γεγονός αυτό θα γίνεται εισαγωγή ενός νέου κόμβου τύπου animal στη λίστα ζώων. Μετά από κάθε εισαγωγή, η λίστα ζώων πρέπει να παραμένει ταξινομημένη. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
L <aid> <eco_id>
    Animals = <aid_1>, <aid_2>, ..., <aid_n>
DONE
```

όπου n είναι ο αριθμός των κόμβων στη λίστα ζώων και για κάθε $i \in \{1, \ldots, n\}$, $<aid_i>$ είναι το αναγνωριστικό του ζώου που αντιστοιχεί στον i-οστό κόμβο της λίστας αυτής.

– **D**

Γεγονός τύπου *distribute animals* το οποίο σηματοδοτεί τον διαχωρισμό των ζώων στα οικοσυστήματα στα οποία ζουν μέσα στο ζωολογικό πάρκο. Μετά την εκτέλεση του γεγονότος αυτού, η λίστα των ζώων είναι κενή. Κατά το γεγονός αυτό πρέπει να κατανεμηθούν όλα τα ζώα της λίστας ζώων στον πίνακα οικοσυστημάτων του ζωολογικού πάρκου. Αυτό μπορεί να γίνει εξετάζοντας το πεδίο eco_id του κάθε κόμβου της λίστας ζώων και στη συνέχεια να γίνεται εισαγωγή του κάθε ζώου στη αντίστοιχη λίστα του οικοσυστήματος στο οποίο ανήκει. Για παράδειγμα αν το πεδίου eco_id ενός κόμβου της λίστας ζώων έχει τιμή 5, το συγκεκριμένο ζώο θα τοποθετηθεί στο οικοσύστημα που βρίσκεται στη θέση 5 του πίνακα οικοσυστημάτων. Επιπλέον μετά από κάθε εισαγωγή η λίστα ζώων του οικοσυστήματος πρέπει να είναι ταξινομημένη με βάση το αναγνωριστικό των ζώων. **Η διαδικασία αυτή θα πρέπει να εκτελείται σε χρόνο O(n), όπου n είναι ο αριθμός των ζώων που περιέχονται στη λίστα ζώων.** Στο τέλος αυτής της διαδικασίας, όλα τα ζώα της λίστας ζώων πρέπει να έχουν διαμοιραστεί στα οικοσυστήματα του πάρκου.

Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
D
    ECOSYSTEMS:
    <ecosystem_1>: <aid_1,1> . . . <aid_1,n1>
    <ecosystem_2>: <aid_2,1> . . . <aid_2,n2>
    ...
   <ecosystem_50>: <aid_50,1> . . . <aid_50,n20>
DONE
```

όπου για κάθε i, $1 \leq i \leq 50$, n_i είναι το μέγεθος της λίστας ζώων του i-οστού οικοσυστήματος του πίνακα

οικοσυστημάτων, και για κάθε j, $1 \leq j \leq n_i$, και <$aid_{i,j}$> είναι το αναγνωριστικό του j-οστού κόμβου στη λίστα ζώων του i-οστού οικοσυστήματος.

– **V <vid> <year>**

Γεγονός τύπου visit το οποίο σηματοδοτεί την επίσκεψη ενός επισκέπτη με αναγνωριστικό <vid> στο ζωολογικό πάρκο το έτος <year>. Κατά το γεγονός αυτό αρχικά θα πρέπει να γίνεται αναζήτηση του επισκέπτη με αναγνωριστικό vid στη λίστα επισκεπτών. Αν αυτός βρεθεί τότε πρέπει να γίνει ανανέωση του πεδίου last_visit με την τιμή <year> και να αυξηθεί ο μετρητής visits. Αν δεν βρεθεί κόμβος με αναγνωριστικό <vid> στη λίστα, τότε πρέπει να εισάγετε έναν νέο κόμβο με αυτό το αναγνωριστικό στη λίστα. Τα πεδία last_visit και visits του νέου κόμβου θα πρέπει να έχουν τιμές <year> και 1, αντίστοιχα.

Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος, το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
V <vid> <year>
      Visitors = <vid₁ : visits₁: last_visit₁>, ..., <vidₙ : visitsₙ: last_visitₙ>
DONE
```

όπου n είναι ο αριθμός των κόμβων στη λίστα επισκεπτών και για κάθε $i \in \{1, \ldots, n\}$, <vid_i> είναι το αναγνωριστικό του επισκέπτη που αντιστοιχεί στον i-οστό κόμβο της λίστας αυτής, ενώ <$visits_i$> και <$last\_visit_i$> είναι ο αριθμός των επισκέψεων του επισκέπτη με αναγνωριστικό <vid_i> στο πάρκο και η χρονολογία της τελευταίας του επίσκεψης, αντίστοιχα.

– **O <years_interval>**

Γεγονός τύπου *delete old visitors* το οποίο σηματοδοτεί τη διαγραφή των επισκεπτών από τη λίστα επισκεπτών για τους οποίους ισχύει ότι η τελευταία επίσκεψή τους στο πάρκο είχε πραγματοποιηθεί πριν από περισσότερα των <years_interval> έτη. Για να υπολογίσετε πόσα χρόνια έχουν περάσει από την τελευταία επίσκεψη αρκεί να γίνει η αφαίρεση από το τρέχον έτος (2016) του έτους της τελευταίας επίσκεψης του επισκέπτη (πεδίο last_visit). Ο συνολικός χρόνος για να πραγματοποιηθούν όλες οι διαγραφές που απαιτεί το γεγονότος αυτό πρέπει να είναι O(n), όπου n ο αριθμός των κόμβων της λίστας επισκεπτών.

Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
O <years_interval>
      Visitors = <vid₁ : visits₁ : last_visit₁>, ..., <vidₙ : visitsₙ : last_visitₙ>
DONE
```

όπου n είναι ο αριθμός των κόμβων στη λίστα επισκεπτών και για κάθε $i \in \{1, \ldots, n\}$, <vid_i>, <$visits_i$> και <$last\_visit_i$> είναι το αναγνωριστικό, ο αριθμός και η χρονολογία της τελευταίας επίσκεψης αντίστοιχα του επισκέπτη που αντιστοιχεί στον i-οστό κόμβο της λίστας αυτής.

– **H <eid>**

Γεγονός τύπου *hire employee* το οποίο σηματοδοτεί την πρόσληψη ενός νέου εργαζόμενου στο πάρκο. Ο εργαζόμενος αρχικά έχει κενή λίστα οικοσυστημάτων για τα οποία είναι υπεύθυνος και συνεπώς ο αντίστοιχος μετρητής (eco_count) έχει τιμή 0. Η εισαγωγή του εργαζομένου στη λίστα πρέπει να πραγματοποιείται σε χρόνο O(1).

Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
H <eid>
    EMPLOYEES: <eid_1>, <eid_2> . . . <eid_n>
DONE
```

όπου n είναι το μέγεθος της λίστας εργαζομένων και για κάθε i, $1 \leq i \leq n$, $<eid_i>$ είναι το αναγνωριστικό του i-οστού εργαζομένου.

– **A <eid> <eco_id>**

Γεγονός τύπου *assign ecosystem σε εργαζόμενο* το οποίο σηματοδοτεί την ανάθεση του οικοσυστήματος με αναγνωριστικό <eco_id> στον εργαζόμενο με αναγνωριστικό <eid>. Το οικοσύστημα με αναγνωριστικό <eco_id> προστίθεται στη λίστα των οικοσυστημάτων για τα οποία ο συγκεκριμένος εργαζόμενος είναι υπεύθυνος. Η εισαγωγή θα πρέπει να πραγματοποιείται σε χρόνο O(n), όπου n είναι το μέγεθος της λίστας εργαζομένων (είναι ανεξάρτητο από το μέγεθος της λίστας οικοσυστημάτων του εκάστοτε εργαζομένου). Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
A <eid> <eco_id>
    EMPLOYEES:
    <employee_1: eco_count_1>: <eco_id_1,1> . . . eco_id_1,n1>
    <employee_2: eco_count_2>: <eco_id_2,1> . . . <eco_id_2,n2>
    ...
   <employee_k: eco_count_k>: <eco_id_k,1> . . . <eco_id_k,nk>
DONE
```

όπου k είναι το μέγεθος της λίστας εργαζομένων, για κάθε i, $1 \leq i \leq k$, n_i είναι το μέγεθος της λίστας οικοσυστημάτων του i-οστού εργαζομένου και για κάθε j, $1 \leq j \leq n_i$, και $<eco\_id_{i,j}>$ είναι το αναγνωριστικό του i-οστού οικοσυστήματος του j-οστού εργαζομένου.

– **R <eid>**

Γεγονός τύπου retirement εργαζομένου το οποίο σηματοδοτεί τη συνταξιοδότηση του εργαζομένου με αναγνωριστικό <eid>. Κατά το γεγονός αυτό πρέπει να διαγράψετε από τη λίστα εργαζομένων τον κόμβο που αντιστοιχεί στον συγκεκριμένο εργαζόμενο. Τα οικοσυστήματα για τα οποία είναι υπεύθυνος ο εργαζόμενος πρέπει να ανατεθούν σε άλλους εργαζομένους ώστε να συνεχιστεί η ομαλή λειτουργία του πάρκου. Πιο συγκεκριμένα αυτά θα ανατίθενται στους εργαζόμενους που βρίσκονται στις τρεις επόμενες και στις τρεις προηγούμενες θέσεις της λίστας εργαζομένων σε σχέση με αυτόν που συνταξιοδοτείται. Για

να το πετύχετε αυτό αρχικά πρέπει να διαιρέσετε τον αριθμό των οικοσυστημάτων που είναι υπεύθυνος ο εργαζόμενος που συνταξιοδοτείται με το εξι (6), αφού τα οικοσυστήματα του θα διανεμηθούν σε 6 άλλους εργαζόμενους (τρείς προηγούμενους και τρείς επόμενους). Το αποτέλεσμα αυτής της πράξης αποτελεί τον αριθμό των οικοσυστημάτων που πρέπει να προστεθούν στη λίστα οικοσυστημάτων του κάθε εργαζομένου που θα αναλάβει μέρος των υποχρεώσεων αυτού που συνταξιοδοτείται. Η ανάθεση γίνεται με τη σειρά που βρίσκονται στη λίστα οικοσυστημάτων του εργαζομένου που αποχωρεί ξεκινώντας απ' τον τρίτο προηγούμενο εργαζόμενου μέχρι τον τρίτο επόμενο στη λίστα εργαζομένων.

Για παράδειγμα αν ο εργαζόμενος που βγαίνει στη σύνταξη είναι υπεύθυνος για 12 οικοσυστήματα με αναγνωριστικά 1,2, …, 12, τότε ο τρίτος προηγούμενος εργαζόμενος θα αναλάβει τα οικοσυστήματα 1,2, ο δεύτερος προηγούμενος τα οικοσυστήματα 3,4 κ.ο.κ. μέχρι να φτάσετε στον τρίτο επόμενο εργαζόμενο, ο οποίος θα αναλάβει τα οικοσυστήματα με αναγνωριστικά 11, 12.

Στην περίπτωση που το αποτέλεσμα της διαίρεσης δεν είναι ακέραιος αριθμός (αλλά είναι μεγαλύτερος από ένα) τότε θα πρέπει να αναθέσετε στους ενεργούς εργαζομένους επιπλέον οικοσυστήματα όσα το ακέραιο μέρος της διαίρεσης. Εξαίρεση αποτελεί ο τελευταίος εργαζόμενος (ο τρίτος επόμενος του εργαζόμενου που συνταξιοδοτείται) ο οποίος θα λάβει όσα οικοσυστήματα έχουν περισσέψει απ' τις προηγούμενες αναθέσεις.

Για παράδειγμα αν ο εργαζόμενος που βγαίνει στη σύνταξη είναι υπεύθυνος για 7 οικοσυστήματα με αναγνωριστικά 1,2, …, 7, τότε ο τρίτος προηγούμενος εργαζόμενος θα αναλάβει το οικοσύστημα 1, ο δεύτερος προηγούμενος το οικοσύστημα 2 κ.ο.κ. μέχρι να φτάσετε στον τρίτο επόμενο εργαζόμενο, ο οποίος θα αναλάβει τα οικοσυστήματα με αναγνωριστικά 6, 7.

Στην περίπτωση που το αποτέλεσμα της διαίρεσης είναι μικρότερο από το ένα τότε δεν θα λάβουν όλοι οι επόμενοι και προηγούμενοι εργαζόμενοι επιπλέον οικοσυστήματα. Ξεκινώντας απ' τον τρίτο προηγούμενο εργαζόμενο θα προστέθετε ένα οικοσύστημα και θα προχωράτε στον επόμενο εργαζόμενο της λίστας μέχρι να έχετε αναθέσει όλα τα οικοσυστήματα που εργαζομένου που αποχωρεί στους υπόλοιπους εργαζομένους σύμφωνα με την παραπάνω περιγραφή.

Για παράδειγμα αν ο εργαζόμενος που βγαίνει στη σύνταξη είναι υπεύθυνος για 2 οικοσυστήματα με αναγνωριστικά 1, 2, τότε ο τρίτος προηγούμενος εργαζόμενος θα αναλάβει το οικοσύστημα 1 και ο δεύτερος προηγούμενος το οικοσύστημα 2. Οι επόμενοι εργαζόμενοι της λίστας δεν λάβουν κανένα επιπλέον οικοσύστημα.

Σημείωση: Δεν χρειάζεται να ελέγχετε αν οι εργαζόμενοι είναι περισσότεροι από 6 καθώς στα test files που θα σας δοθούν εξασφαλίζεται ότι οι «ενεργοί» εργαζόμενοι είναι περισσότεροι από 6.

Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
R <eid>
    EMPLOYEES:
    <employee1: eco_count1>: <eco_id1,1> . . . eco_id1,n1>
    <employee2: eco_count2>: <eco_id2,1> . . . <eco_id2,n2>
    ...
   <employeek: eco_countk>: <eco_idk,1> . . . <eco_idk,nk>
DONE
```

όπου k είναι το μέγεθος της λίστας εργαζομένων, για κάθε i, $1 \leq i \leq k$, n_i είναι το μέγεθος της λίστας οικοσυστημάτων του i-οστού εργαζομένου και για κάθε j, $1 \leq j \leq n_i$, και <eco_id$_{i,j}$> είναι το αναγνωριστικό του i-οστού οικοσυστήματος του j-οστού εργαζομένου.

– G

Γεγονός τύπου gold visitors κατά το οποίο υπολογίζονται και τυπώνονται οι πέντε (5) πιο συχνοί επισκέπτες του πάρκου εξετάζοντας το πεδίο visits των κόμβων της λίστας επισκεπτών. Για το σκοπό αυτό θα χρησιμοποιήσετε έναν βοηθητικό πίνακα 5 θέσεων στον οποίο θα αποθηκεύονται δείκτες προς τους κόμβους της λίστας επισκεπτών.

Πιο συγκεκριμένα θα εφαρμόσετε τις εξής ενέργειες:

i. Ο βοηθητικός πίνακας αρχικοποιείται με τους 5 πρώτους εργαζομένους της λίστας εργαζομένων.
ii. Καθώς εκτελείται η διάσχιση της λίστας εργαζομένων, για κάθε κόμβο που επισκέπτεστε, εξετάζετε αν ο μετρητής visits έχει τιμή μεγαλύτερη από τον εργαζόμενο με τις λιγότερες επισκέψεις στον βοηθητικό πίνακα. Αν συμβαίνει αυτό, τότε αντικαθιστούμε τον εργαζόμενο αυτό στον πίνακα με τον τρέχον κόμβο της λίστας. Στην συνέχεια, θα πρέπει να υπολογίζετε εκ νέου ποιος είναι ο επισκέπτης με τις λιγότερες επισκέψεις στον πίνακα και να επαναλαμβάνετε.
iii. Μετά το τέλος της διάσχισης της λίστας εργαζομένων, ο βοηθητικός πίνακας θα περιέχει δείκτες στους κόμβους της λίστας που αντιστοιχούν στους 5 πιο συχνούς επισκέπτες του πάρκου.

Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
G
    Gold Visitors = <vid1 : visits1: last_visit1>, ..., <vid5 : visits5: last_visit5>
DONE
```

και για κάθε $i \in \{1, \ldots, 5\}$, <vid$_i$>, <visits$_i$> και <last_visit$_i$> είναι το αναγνωριστικό, ο αριθμός των επισκλεψεων και η χρονολογία της τελευταίας επίσκεψης αντίστοιχα του επισκέπτη που αντιστοιχεί στην i-οστή θέση του βοηθητικού πίνακα.

– X

Γεγονός τύπου print ecosystems το οποίο σηματοδοτεί την εκτύπωση όλων των οικοσυστημάτων του πίνακα οικοσυστημάτων. Για το κάθε οικοσύστημα θα πρέπει να εκτυπώνονται όλα τα στοιχεία του, συμπεριλαμβανομένων της λίστας ζώων που κατοικούν σ' αυτό. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
X
    ECOSYSTEMS:
    <ecosystem1: animal_count1>: <aid1,1> . . . <aid1,n1>
    <ecosystem2: animal_count2>: <aid2,1> . . . <aid2,n2>
    ...
   <ecosystem50: animal_count50>: <aid50,1> . . . <aid50,n50>
```

όπου για κάθε i, $1 \leq i \leq 50$, n_i είναι το μέγεθος της λίστας ζώων του i-οστού οικοσυστήματος του πίνακα οικοσυστημάτων, και για κάθε j, $1 \leq j \leq n_i$, και $<aid_{i,j}>$ είναι το αναγνωριστικό του j-οστού κόμβου στη λίστα ζώων του i-οστού οικοσυστήματος.

– **Y**

Γεγονός τύπου print visitors το οποίο σηματοδοτεί εκτύπωση όλων των επισκεπτών της λίστας επισκεπτών. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
Y
      Visitors = <vid1 : visits1: last_visit1> ..., <vidn : visitsn : last_visitn>
DONE
```

όπου n είναι ο αριθμός των κόμβων στη λίστα επισκεπτών και για κάθε $i \in \{1, \ldots, n\}$, $<vid_i>$ είναι το αναγνωριστικό του επισκέπτη που αντιστοιχεί στον i-οστό κόμβο της λίστας αυτής.

– **Z**

Γεγονός τύπου print employees το οποίο σηματοδοτεί εκτύπωση όλων των κόμβων της λίστας εργαζομένων. Για τον κάθε εργαζόμενο θα πρέπει να εκτυπώνονται όλα τα στοιχεία του, συμπεριλαμβανομένων της λίστας οικοσυστημάτων για τα οποία αυτός είναι υπεύθυνος. Μετά το πέρας της εκτέλεσης ενός τέτοιου γεγονότος το πρόγραμμα θα πρέπει να τυπώνει την ακόλουθη πληροφορία:

```
Z
    EMPLOYEES:
    <employee1: eco_count1>: <eco_id1,1> . . . eco_id1,n1>
    <employee2: eco_count2>: <eco_id2,1> . . . <eco_id2,n2>
    ...
   <employeek: eco_countk>: <eco_idk,1> . . . <eco_idk,nk>
DONE
```

όπου k είναι το μέγεθος της λίστας εργαζομένων, για κάθε i, $1 \leq i \leq k$, n_i είναι το μέγεθος της λίστας οικοσυστημάτων του i-οστού εργαζομένου και για κάθε j, $1 \leq j \leq n_i$, $<eco\_id_{i,j}>$ είναι το αναγνωριστικό του i-οστού οικοσυστήματος του j-οστού εργαζομένου.

Δομές Δεδομένων

Στην υλοποίησή σας δεν επιτρέπεται να χρησιμοποιήσετε έτοιμες δομές δεδομένων (πχ., ArrayList) είτε η υλοποίηση πραγματοποιηθεί στη C είτε στη Java. Στη συνέχεια παρουσιάζονται οι δομές σε C που πρέπει να χρησιμοποιηθούν για την υλοποίηση της παρούσας εργασίας.

```
struct animal {
	int aid;
	int eco_id;
	struct animal *next;
};

struct ecosystem {
	int eco_id;
	int animal_count;
	struct animal * animals_list;
};

struct visitor {
	int vid;
	int last_visit;
	int visits;
	struct visitor * next;
};

struct employee_eco {
	int eco_id;
	struct employee_eco * next;
};

struct employee {
	int eid;
	int eco_count;
	struct employee_eco * employee_ecos_list;
	struct employee * prev;
	struct employee * next;
};
```

```
/*The array of the park's ecosystems */
struct ecosystem ecosystems_array [50];

/* global variable, pointer to the beginning of the animals list*/
struct animal * animals_list;

/* global variable, pointer to the beginning of the visitors list*/
struct visitor * visitors_list;

/* global variable, pointer to the sentinel node of the users list */
struct employee * employees_sentinel;
```